Poincaré-Birkhoff-Witt代数のグレブナー基底計算とRisa/Asirへの実装 (数式処理とその周辺分野の研究)

AUTHOR(S):

小原, 功任; 田島, 慎一

CITATION:

小原, 功任 ...[et al]. Poincaré-Birkhoff-Witt代数のグレブナー基底計算とRisa/Asirへの実装 (数式処理とその周辺分野の研究). 数理解析研究所講究録 2017, 2054: 134-138

ISSUE DATE:

2017-10

URL:

http://hdl.handle.net/2433/237154

数理解析研究所講究録
第2054巻 2017年 134-138

Poincaré-Birkhoff-Witt代数のグレブナー基底計算とRisa/Asir への実装

小原功任 *
KATSUYOSHI OHARA
金沢大学理工研究域
FACULTY OF MATHEMATICS AND PHYSICS,
INSTITUTE OF SCIENCE AND ENGINEERING,
KANAZAWA UNIVERSITY

田島慎一 †
SHINICHI TAJIMA
筑波大学数理物質系
INSTITUTE OF MATHEMATICS,
FACULTY OF PURE AND APPLIED SCIENCES,
UNIVERSITY OF TSUKUBA

1 Poincaré-Birkhoff-Witt 代数

われわれは，計算機代数システム Risa/Asir に，左 Poincaré-Birkhoff-Witt イデアルのグレブナー基底を計算するアルゴリズムを実装した．本稿ではわれわれの実装について報告する．

まず多項式環の概念を拡張して，左 Poincaré-Birkhoff-Witt 代数を導入し，その環上における左イデアルのグレブナー基底とその計算法について概説する．左 Poincaré-Birkhoff-Witt 代数上のグレブナー基底に関する理論の詳細については，文献 [1] の第 2 章を参照されたい．

K を体とし，非可換な不定元 $x_1,\dots,x_n$ を考える．ただし，K の元と各 x_i は可換である．非負整数ベクトル $\alpha=(\alpha_1,\dots,\alpha_n)\in\mathbf{N}_0^n$ に対応する不定元の巾積を

$$x^\alpha=\overbrace{x_1\cdots x_1}^{\alpha_1}\overbrace{x_2\cdots x_2}^{\alpha_2}\cdots\overbrace{x_n\cdots x_n}^{\alpha_n}$$

で定める．乗法が非可換であるので，一般に $x^{(2,2)}=x_1x_1x_2x_2\neq x_2x_2x_1x_1$ である．

環 R を K-ベクトル空間とみたとき，単項式集合 $\mathcal{M}=\{x^\alpha\mid\alpha\in\mathbf{N}_0^n\}$ が基底をなすならば，R を K 上の**左多項式環**という．つまり，R の零でない元 f は，

$$f=\sum_{\alpha\in\mathbf{N}_0^n}c_\alpha x^\alpha$$

と一意に書けて (標準形)，かつ $\mathcal{N}(f)=\{\alpha\in\mathbf{N}_0^n\mid c_\alpha\neq 0\}$ は有限集合となる．このとき，通常の多項式環のグレブナー理論と同様に，単項式集合 $\mathcal{M}$ の指数部分は加法的半群をなし，いわゆる**単項式順序** (admissible order) の概念が定まる．また，単項式順序 $\prec$ が与えられたとき，$\exp(f)=\max_\prec\mathcal{N}(f)$ と表すこととする．また，$\mathrm{in}_\prec(f)=c_{\exp(f)}x^{\exp(f)}$ を f の**イニシャル単項式**と呼ぶ．

いま，単項式順序 $\prec$ をひとつ固定する．

定義 1. 左多項式環 R が，**左 Poincaré-Birkhoff-Witt 代数**であるとは，単項式順序 $\prec$ が存在して，任意の $f,g\in R$ に対して，$\exp(fg)=\exp(f)+\exp(g)$ を満すことである．

*ohara@se.kanazawa-u.ac.jp
†tajima@math.tsukuba.ac.jp

この定義から分かるように左 Poincaré-Birkhoff-Witt 代数においては，積の順序交換によって，$\exp(\cdot)$ は不変である．ふたつの単項式 x_ix_j と x_jx_i の $\exp(\cdot)$ が等しいことに注意すると，ある $q_{ji} \in K$ が存在して，多項式 $p_{ji} = x_jx_i - q_{ji}x_ix_j$ が，$\exp(p_{ji}) \prec \exp(x_ix_j)$ を満すようにできる．すなわち，関係式

$$x_jx_i = q_{ji}x_ix_j + p_{ji} \quad \text{かつ} \quad \exp(p_{ji}) \prec \exp(x_ix_j)$$

が成り立つ．逆に，n 次正方行列 $Q = (q_{ji})_{ji} \in M_n(K)$ および $P = (p_{ji})_{ji} \in M_n(R)$ を考えると，組 $\mathcal{R} = (Q, P)$ が，左 Poincaré-Birkhoff-Witt 代数の交換関係を定めている．このことを，$R = K\langle x; \mathcal{R}\rangle$ と表す．

例 1. 可換な不定元 $x_1, \ldots, x_n$ に対して，多項式環 $K[x_1, \ldots, x_n]$ は明らかに左 Poincaré-Birkhoff-Witt 代数を与える．

例 2. Weyl 代数 $D_n = K\langle x_1, \ldots, x_n, \partial_1, \ldots, \partial_n\rangle$ は，$x_{n+i} = \partial_i$ と置けば，交換関係

$$x_jx_i = x_ix_j + \delta_{j,i+n}, \quad (1 \le i < j \le 2n)$$

を満す．したがって，$\mathcal{R} = \{(1, \delta_{j,i+n}) \mid 1 \le i < j \le 2n\}$ とするとき，$D_n = K\langle x_1, \ldots, x_{2n}; \mathcal{R}\rangle$ が成り立つ．

左 Poincaré-Birkhoff-Witt 代数における左イデアルのグレブナー基底を，多項式環のときと同様に定義しよう．

定義 2. 左 Poincaré-Birkhoff-Witt 代数 R の左イデアル $I \neq \{0\}$ に対して，有限集合 $G \subset I \setminus \{0\}$ が I のグレブナー基底であるとは，モノイデアル $\mathrm{Exp}(I) = \{\exp(f) \mid f \in I,\ f \neq 0\}$ が，$\{\exp(g) \mid g \in G\}$ で生成されることである．

さて，左 Poincaré-Birkhoff-Witt 代数の定義から分かるように，ふたつの元 $f, g \in R$ が与えられたとき，$\exp(fg) = \exp(gf)$ が成り立つ．よって，適当な定数をかけることで，fg と gf の，それぞれのイニシャル単項式は互いに打ち消しあう．つまり，$fg - cgf$ に現れるすべての単項式が $\mathrm{in}_\prec(fg)$ よりも小さくすることができる．よって，次のように S 多項式を定義すればよい．

命題 1. 左 Poincaré-Birkhoff-Witt 代数の零でない元 f, g に対して，$\alpha = \exp(f), \beta = \exp(g), \gamma = \mathrm{lcm}(\alpha, \beta)$ とおく．このとき，

$$S(f,g) = \frac{x^{\gamma-\alpha}}{\mathrm{lc}_\prec(x^{\gamma-\alpha}f)} f - \frac{x^{\gamma-\beta}}{\mathrm{lc}_\prec(x^{\gamma-\beta}g)} g$$

を f, g の **S 多項式**という．ここで，$\mathrm{lc}_\prec(\cdot)$ はイニシャル単項式の係数である．

左 Poincaré-Birkhoff-Witt 代数の定義から，単項簡約によって順序の高い項が新たに出現することはない．したがって多項式環の場合と同様に，左 Poincaré-Birkhoff-Witt 代数でもブッフバーガーのアルゴリズムが働き，グレブナー基底を計算することが可能である．われわれは，実際に，左 Poincaré-Birkhoff-Witt 代数における算術演算と，ブッフバーガーのアルゴリズムを計算機代数システム Risa/Asir に実装した．その他，計算機代数システムにおける既存の実装としては，Levandovskyy による Singular への実装 [2] がある．

2 Risa/Asir への実装 (一般の場合)

まず 左 Poincaré-Birkhoff-Witt 代数は左多項式環であるから，零でない元は，標準形 $f = \sum_{\alpha \in \mathbf{N}_0^n} c_\alpha x^\alpha$ で表すことができる．われわれの実装では 左 Poincaré-Birkhoff-Witt 代数の元は，標準形で表示すると約

束する．零元は記号 0 で表す．このことから，左 Poincaré-Birkhoff-Witt 代数における加法は，標準形を通常の多項式と同様に扱って，和を計算すればよいことが分かる．したがって，問題になるのは積の計算である．

積の計算には，R における交換関係が必要になる．前節で述べたように，左 Poincaré-Birkhoff-Witt 代数 $R = K\langle x; \mathcal{R}\rangle$ は，関係式を表す $\mathcal{R} = (Q, P)$ によって定まるから，$\mathcal{R}$ を環の定義として与えることができる．われわれの実装では，関数 `yw.define_ring` を用いて環を定義する．具体的には以下のような引数をとる．

```
yw.define_ring(["pbw", X, Q, P])
```

X は変数集合のリストであり，Q, P は $\mathcal{R}$ の各成分である．ただし，われわれの実装では，行列 Q, P は対角成分よりも下の部分しか用いられない．したがって，P, Q は対角成分が 0 の下三角行列として与えてもよい．環が定義されているとき，関数 `yw.mul` によって乗算を実行し，結果を標準形で返す．交換関係 $\mathcal{R}$ から直接的には結果を得られない場合は，交換関係 $\mathcal{R}$ を繰り返し適用して，標準形を求める．

例 3. 2 変数の 左 Poincaré-Birkhoff-Witt 代数 $R = K\langle x, y; \mathcal{R}\rangle$ を以下の交換関係で定める．

$$yx = xy - y + x$$

このとき，$f = y^2 + 1$, $g = xy - 1$ の積 fg を計算するには，次のように入力する．

```
[0] import("yw.rr")$
[1] Q=newmat(2,2,[[0,0],[1,0]]);
[2] P=newmat(2,2,[[0,0],[y-x,0]]);
[3] yw.define_ring(["pbw", [x,y], Q, P] );
[4] F=y^2+1;
[5] G=x*y-1;
[6] yw.mul(F,G);
(y^3-2*y^2+2*y)*x+2*y^3-2*y^2-1
```

この結果は，$fg = x(y^3 - 2y^2 + 2y) + 2y^3 - 2y^2 - 1$ を意味する．

グレブナー基底計算においては，各元は Risa/Asir の組み込み型である分散表現多項式で表されている．よって，Risa/Asir の組み込み関数 `dp_ord` によって，単項式順序を定義することができる．

例 4. 先の例で述べた，$R = K\langle x, y; \mathcal{R}\rangle$ に対し，重みベクトル $w = (1, 2)$ と置く．単項式 $x^{a_1}y^{b_1}$ と $x^{a_2}y^{b_2}$ を比較するとき，まず内積 $\langle w, (a_i, b_i)\rangle = a_i + 2b_i$ で比較し，tie braker として，全次数逆辞書式順序を用いるような単項式順序を定義するには，以下のように入力すればよい．

```
[7] dp_ord(|v=[x,y],order=[[x,1,y,2],[@grlex,range(x,y)]]);
```

この単項式順序のもとで，例 3 で与えられた 左 Poincaré-Birkhoff-Witt 代数の元 fg および f^2 の生成する左イデアルの簡約グレブナー基底を求めよう．

例 5.

```
[8] L=[yw.mul(F,G), yw.mul(F,F)];
[9] Gr=yw.gr(L);
[x-y,-x^2-1]
```

この結果は $G = \{x - y, -x^2 - 1\}$ が簡約グレブナー基底であることを意味する．

3 Risa/Asir への実装 (特別な場合)

ここまでに述べた方法では，左 Poincaré-Birkhoff-Witt 代数における乗算は，交換関係 $\mathcal{R}$ を繰り返し適用することで実現するものであった．数学ソフトウェアを書いた経験があればすぐ分かることであるが，パターンの再帰的適用には大きな計算時間を要し，効率的とはいえないことが多い．したがって，よく用いられる交換関係に対しては，再帰的なルールの適用を避けるような特別な実装が施されていることが望ましい．

われわれの実装では，以下のタイプの交換関係のみを含む場合を特別に扱う．

(1) $yx = xy + 1$

(2) $yx = xy + x$

タイプ (1) は，Weyl 代数における交換関係に相当する．このとき，変数 y は変数 x に対する微分作用素 ∂_x とみなすことができるのはよく知られている．したがって，いわゆる積の微分公式から，

$$y^i x^j = \sum_{k=0}^{\min(i,j)} \binom{i}{k}\binom{j}{k} k!\, x^{j-k} y^{i-k}$$

なる交換関係が得られる．この公式を用いて，二つの左多項式の積を左多項式に書き換えることができる．

一方，タイプ (2) における変数 x, y の関係は，変数 y を変数 x に対するオイラー微分作用素 $\theta_x = x\partial_x$ とみなした場合の交換関係と同じである．よく知られているように，$\theta_x x^m = x^m(\theta_x + m)$ と書けるから，したがって，

$$y^i x^j = x^j (y+j)^i = \sum_{k=0}^{i} \binom{i}{k} j^{i-k} x^j y^k$$

なる交換関係を得る．

われわれの実装では，タイプ (1) の交換関係を記号 `weyl` または `partial` で，タイプ (2) を記号 `euler` で表す．したがって，交換関係

$$ts = st + s, \quad yx = xy + 1, \quad vu = uv + 1 \tag{1}$$

をみたす 左 Poincaré-Birkhoff-Witt 代数 $R = K\langle s, t, x, y, u, v; \mathcal{R}\rangle$ を定義するには，

```
[10] yw.define_ring(["euler", [s], "weyl", [x,u]]);
```

と入力する．シンボル `ds`, `dx`, `du` でそれぞれ，変数 t, y, v が表される．この形式の定義が行われた場合，自動的に高速な乗算プログラムが使用される．

例 6. 交換関係 (1) で定まる 左 Poincaré-Birkhoff-Witt 代数 $R = K\langle s, t, x, y, u, v; \mathcal{R}\rangle$ における多項式

$$f_1 = xu^2 s + t, \quad f_2 = u^2 s + y, \quad f_3 = 2xus + v$$

を考える．このとき積 $f_1 f_2$ は

$$f_1 f_2 = (t + su^2 x)y + su^2 t + s^2 u^4 x + su^2$$

となる．R における単項式順序を，ブロック順序 $s > x, u, t, y, v$ かつ変数 x, u, t, y, v に対しては，全次数逆辞書式順序を適用するものとしよう．このとき，左イデアル $\langle f_1, f_2, f_3\rangle$ ののでのグレブナー基底は，

$$G = \{-uv+2t,\ -xy+t,\ -v^2y-4st^2-6st-2s,\ -vy-2sut-2su,\ -v^2-4sxt-2sx,\ y+su^2,\ v+2sux\}$$

となる．Risa/Asir への入力は以下の通りである．

```
[11] yw.define_ring(["euler",[s],"weyl",[x,u]]);
{[euler,[s],weyl,[x,u]],[s,x,u],[0,3,3],[0,0,0],[ds,dx,du],[s,x,u,ds,dx,du]}
[12] L = [x*u^2*s + ds, u^2*s + dx, 2*x*u*s + du]$
[13] V=yw.ring_vars()$
[14] dp_ord(|v=V,order=[[s,1],[@grlex,range(x,du)]])$
[15] Gr=yw.gr(L);
[-u*du+2*ds,-x*dx+ds,-du^2*dx-4*s*ds^2-6*s*ds-2*s,-du*dx-2*s*u*ds-2*s*u,
-du^2-4*s*x*ds-2*s*x,dx+s*u^2,du+2*s*u*x]
```

参 考 文 献

[1] J. Bueso, J. Gomez-Torrecillas and A. Verschoren, Algorithmic Methods in Non-Commutative Algebra, Springer, 2003.

[2] V. Levandovskyy and H. Schönemann, Plural — a computer algebra system for noncommutative polynomial algebras, Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation (ISSAC '03), 176–183, ACM Press, 2003.